Accessibility

Accessibility means that users can use your site to get to your information. That's all. It means that you don't put up any barriers or impediments to getting to your information.

It doesn't mean that your web site is dull and colorless. It doesn't mean you can't have audio and video files. It doesn't mean you have to buy expensive software or hire experts to test each page of your site.

Accessibility just means that you don't code your pages in a way that keeps people from getting to your information.

The good news is that, if you have followed the standards and guidelines in this guide so far, you are eighty to ninety percent accessible already. We'll show you how to get the other ten or twenty percent in this section.

508 Accessibility Standard

The 508 accessibility standards come from the federal Rehabilitation Act Amendments of 1998, which requires the U.S. federal government to make sure that any electronic or information technology they purchase will allow disabled citizens and employees to access or use the technology. Many states, including Missouri, have adopted these standards in state law.

While Section 508 covers everything from copiers to software, we are concerned with how it affects web pages.

WAI Standard

The other accessibility standard comes from the W3C, and is called the Web Accessibility Initiative, or WAI for short. It features a three-tiered approach to accessibility, with the first tier being most important, the second tier a little less important, and the third tier relegated to “nice to have” status.

You can think of the WAI standard as the official standard of the web, and as close to an international standard as we can get.

The Missouri Standard

Missouri has adopted the 508 standard in Section 191.863 of the Revised Statutes, which states:

> “1. The [Assistive technology] council shall work in conjunction with the office of information technology to assure state compliance with the provisions of Section 508 of the Workforce Investment Act of 1998 regarding accessibility of information technology for individuals with disabilities.
>
> 2. When developing, procuring, maintaining or using information technology, each state department or agency shall ensure, unless an undue burden would be imposed on the department or agency, that the information technology allows employees, program participants and members of the general public access to and

use of information and data that is comparable to the access by individuals without disabilities. "

OIT has developed a standard to implement the statute. You can download the standard at: (http://oit.mo.gov/standards/ITGS0003_Missouri_IT_Accessibility_Standards.doc)

The Missouri standards closely follow the 508 standards on web development, with the exceptions of (b) and (p). The Missouri standards read as:

Web-based Intranet and Internet Information and Applications

(a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content) except for captioning of audio information which shall comply with (b) of this section.

(b) Captioning, video description or other equivalent alternatives for multimedia presentations, excluding live Webcasts, shall be provided in synchrony with the presentation, and in accordance with the following:

(i) Captioning shall be provided for multimedia presentations that contain speech or other audio information necessary for the comprehension of the content in accordance with the schedule established in Paragraph (c) under Video and Multimedia Products.

(ii) Video description shall be provided for multimedia presentations that contain visual information necessary for the comprehension of the content, in accordance with the schedule established in Paragraph (d) under Video and Multimedia Products.

(iii) Live Webcasts that contain speech or other audio information necessary for the comprehension of the content, shall be captioned in accordance with the following schedule with priority given to content of statewide importance and events that do not provide the opportunity to request individual accommodations.

_ By June 30, 2005, 10 percent of all live Webcast hours transmitted during the preceding twelve months by a State department or agency must incorporate captioning.

_ By June 30, 2007, 25 percent of all live Webcast hours transmitted during the preceding twelve months by a State department or agency must incorporate captioning.

_ By June 30, 2009, 50 percent of all live Webcast hours transmitted during the preceding twelve months by a State department or agency must incorporate captioning.

(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

(d) Documents shall be organized so they are readable without requiring an associated style sheet.

(e) Redundant text links shall be provided for each active region of a server-side image map.

(f) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

(g)Row and column headers shall be identified for data tables.

(h)Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.

(i) Frames shall be titled with text that facilitates frame identification and navigation.

(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

(k)A text-only page, with equivalent information or functionality, shall be provided to make a Web site comply with the provisions of these standards, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.

(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.

(m) When a Web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with the standards set forth under “Software Applications and Operating Systems”, paragraphs (a) through (l) contained within this document.

(n)When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

(o)A method shall be provided that permits users to skip repetitive navigation links.

(p)When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

(q)Contact information for issues related to accessibility shall be provided on each entry page.

Item (b) goes into much greater detail about making multimedia accessible and provides a time line for compliance than 508 does..

Item (p) is a new item not found in 508, and calls for an accessibility contact link on each home page.

Missouri Accessibility Implementation

Ok, so how do you implement these standards?

If you've followed the guidelines in the previous sections, you have done the hardest part towards accessibility, and you'll find that you've implemented most of these already.

We will assume that you are using either Internet Explorer 6 or Mozilla Firefox to check your pages. You will need to download either the AIS Web Accessibility Toolbar for IE (http://www.nils.org.au/ais/), or the Web Developers Toolbar for Firefox/Mozilla (http://www.chrispederick.com/work/firefox/webdeveloper/) and Checky (http://checky.sourceforge.net/#Install). If you haven't already installed these on your browsers, do it now. We'll wait. You'll thank us later.

Ready? Let's go through the standards, one by one, and figure out how to hit them.

(a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content) except for captioning of audio information which shall comply with (b) of this section.

Alt-text Guidelines:

1. Keep wording simple
   Bad: alt="DMD's logo and link to DMD home page"
   Good: alt="DMD home"
2. Describe function of graphic rather than what it is or looks like.
   Bad: alt="right-pointed triangle graphic"
   Good: alt="next"
3. Include punctuation or spaces at the end of the alt-text of images used as links. Remember that screen readers look for punctuation to signal pauses and ends of sentences.
4. Of course, if you have been redesigning your site with standards, you won't need any spacer gifs because your site is laid out with CSS, but if you need to use a "spacer.gif" for layout, the alternate text should be alt="" so screen readers don't waste time trying to interpret the text.
5. If you use graphics for buttons or links, the alt text should indicate where you are linking to and not the word "Link" or "button." There is no need to use Link attributes since the href and alt attributes are both used.
   Bad: <a href="#"><img src="button.gif" alt="link" /></a>
   Good: <a href="#"> <img src="button.gif" alt="Digital Media Developers Group" /></a>

Longdesc

A longdesc attribute is a link to a file that provides more information about the image.

Title

A title attribute can be applied to nearly any tag, including the <img> tag. It just gives the user a caption to go with the element. In visual browsers, the title attribute usually shows up as a "tool tip" pop-up message. In audio browsers, the title may just be read aloud. This attribute can be used to provide additional information to an image.

“[D]” link

Since not all browsers can render the longdesc attribute, it is helpful to add a link immediately after the image using “[D]” to link to the description page.

Code example:

```
<img src=”image.jpg” alt=”Chart showing 2004 employment figures”
longdesc=”2004employment.htm” title=”Chart showing 2004 employment figures”
/><a href=”2004employment.htm”>[D]</a>
```

where 2004employment.htm reads:

```
<table class="cspace">
<caption>Missouri Annual Average Employment Statistics</caption>
<thead>
 <tr>
   <th scope=”col”>Year</th>
   <th scope=”col”>Labor Force</th>
            <th scope=”col”>Employment</th>
 </tr>
 </thead>
 <tbody>
 <tr>
 <th  scope=”row”>2003</th>
 <td>3,020,592</td>
 <td>2,850,466<</td>
 </tr>
 </tbody>
 </table>
```

Testing for alt text

- Use your toolbar to turn off images. Do you see any text where the images were?
- Mouse over the images. You should see a “tooltip” in Firefox if there is a title present.

(b) Captioning, video description or other equivalent alternatives for multimedia presentations, excluding live Webcasts, shall be provided in synchrony with the presentation, and in accordance with the following:

i. Captioning shall be provided for multimedia presentations that contain speech or other audio information necessary for the comprehension of the content in accordance with the schedule established in Paragraph (c) under Video and Multimedia Products.

ii. Video description shall be provided for multimedia presentations that contain visual information necessary for the comprehension of the content, in accordance with the schedule established in Paragraph (d) under Video and Multimedia Products.

iii. Live Webcasts that contain speech or other audio information necessary for the comprehension of the content, shall be captioned in accordance with the following schedule with priority given to content of statewide importance and events that do not provide the opportunity to request individual accommodations.

_ By June 30, 2005, 10 percent of all live Webcast hours transmitted during the preceding twelve months by a State department or agency must incorporate captioning.

_ By June 30, 2007, 25 percent of all live Webcast hours transmitted during the preceding twelve months by a State department or agency must incorporate captioning.

_ By June 30, 2009, 50 percent of all live Webcast hours transmitted during the preceding twelve months by a State department or agency must incorporate captioning.

This is one case where the Missouri standard goes beyond the 508 standard by giving a time line for accessible live webcasts. So how do we do this?

- For audio webcasts, make sure there is a transcription available, either from a script or by close-captioning the webcast as it airs.
- For video web casts, provide accessible close captioning of the audio as well as a description of any activity that a visually-impaired user may need to know.
- Place an anchor to a page with a text transcript or description of the clip right next to the anchor for the clip.
- If the user has requested a text-only page, replace all URL references to the clip with URL references to the text transcript or description.

(c)Web pages shall be designed so that all information conveyed with color is also available without color, for example, from context or markup.

There's a couple of aspects to this standard.

First, avoid using words like “click the green button to continue. Click the red button to reset the form.” A color-blind person may only see a couple of gray buttons. Instead, label each button “continue” and “reset” so there is a textual clue to follow.

Second, make sure background patterns and colors contrast well with the text color and select colors that will make pages easy to read by people with color blindness.

Testing for color issues

- Go to http://colorfilter.wickline.org/ or http://www.vischeck.com/vischeck/vischeckURL.php and enter your URL.
- In IE, use the “colour” tab to check your page in grayscale.
- In Firefox, right-click on the page. Select “Checky>>WAI/508>>Color Blind Web Page Filter. Select a type of color blindness to see your page rendered as a color-blind person may see it.

(d)Documents shall be organized so they are readable without

requiring an associated style sheet.

If you've started your web page as a plain text page, then applied styles to add colors, images, fonts and positioning, you won't have much of a problem here.

Testing

Turn off style sheets or view the page in a text browser to see how it displays.

(e)Redundant text links shall be provided for each active region of a server-side image map.

(f)Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

Ah, image maps. They let you point to various parts of an image and click a link. They are a great way to carve up a map of Missouri into regions, counties, cities or anything else you can think of.

Let's set a few rules for them.

1) Server-side image maps should be avoided. Back in the early days of the Internet, browsers did not have the capability of interpreting an image map, so all the link coordinates had to be generated server-side. Naturally, there was little thought of accessibility. When a user mouses over the image map, they get a set of coordinates, but not an actual link that they can see in the status bar. Plus, users have to use a mouse; there is no natural text equivalent for the link.
   If you have to use a server-side image map, you will need to include a separate text list of links on the page to give the users another way to access the information.
2) Client-side image maps are part of a web page and are interpreted by nearly all browsers. They are easy to create and make accessible. There is virtually no kind of shape you cannot make with a client-side map. If you have to use an image map, use this kind.
3) Only use an image map when there is no suitable text-based solution. For instance, to let users pick a region of Missouri on a map. Don't use it to make your navigation bar at the top of your page; that is what text and CSS are for.

Image map hot spots

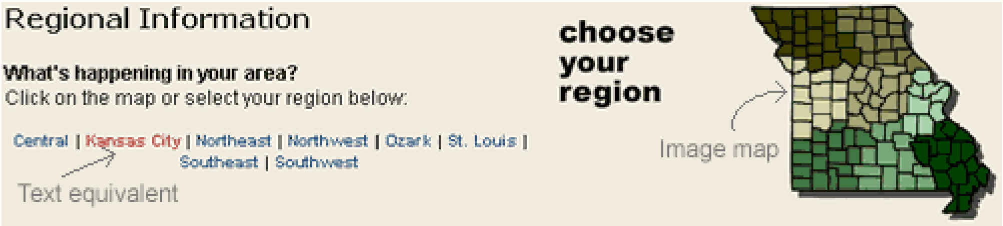
If you have image maps, your main image should have an alt attribute. You also need an alt attribute for each hot spot link.

Example:

```
<img id="statemap" src="countymap.gif" width="557" height="484"
usemap="#countymap" alt="Map of Missouri counties"/>

<map id="countymap">...

<area shape="poly"
```

```
coords="188,221,188,217,143,216,143,227,143,251,144,252,187,252,188,221"
href="/cgi-bin/hoAvail.cgi?sr=1&amp;cnty=83"  title="Henry  County" alt="Henry
County" />...

</map>
```

It is also a good idea to have a text list of links for your image map. For instance, if you have five regions in the state, show the text equivalent somewhere on the page.

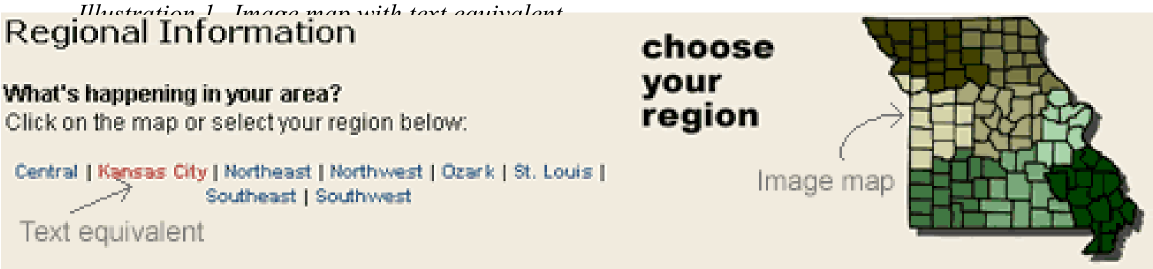


Regional Information

What's happening in your area?
Click on the map or select your region below:

Central | Kansas City | Northeast | Northwest | Ozark | St. Louis | Southeast | Southwest

Illustration 1 Image map with text equivalent



Regional Information

What's happening in your area?
Click on the map or select your region below:

Central | Kansas City | Northeast | Northwest | Ozark | St. Louis | Southeast | Southwest

Testing

Turn off images. Try to navigate.

(g)Row and column headers shall be identified for data tables.

(h)Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.

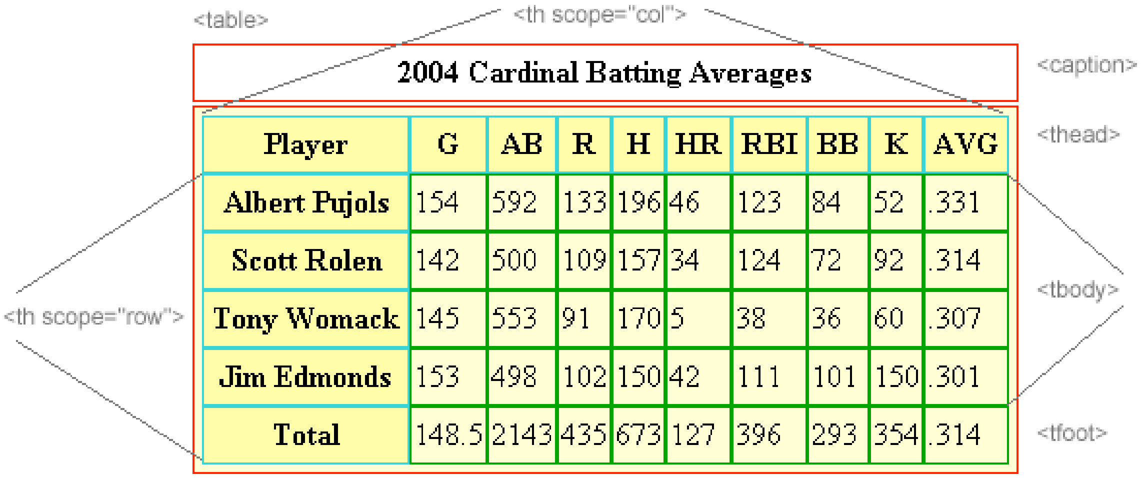
Data tables are one of those things that you really have to work at to make accessible. Fortunately, once you get the hang of it, it doesn't take too long to mark the table up to make it accessible to any modern browser.

There's several things to keep in mind with any table:

- We are talking tables for data here, not layout tables. You may end up with a layout table in your site, but we hope you'll try to avoid nested-table layouts. Try to limit your use of tables for layout by using text and CSS. This is not always possible, but dare to dream.

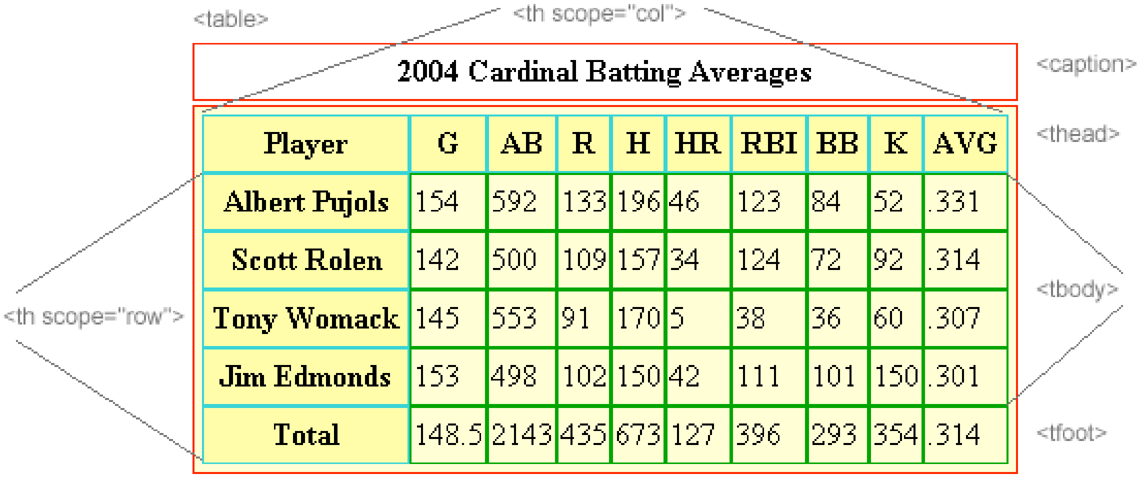
- A data table should be marked up semantically to identify its various parts. This is half the battle of accessibility.
- This standard is the bare minimum for table accessibility. We'll show you how to make it even more accessible.
- If you've tried everything else (or are just lazy or didn't bother to read the rest of this guide) and have had to resort to a layout table, very little of this section is going to apply to that layout table. You should keep your tables very simple, since you *don't want* the layout tables to convey any information.

Let's take a look at a basic data table in Illustration 2:



2004 Cardinal Batting Averages

Player	G	AB	R	H	HR	RBI	BB	K	AVG
Albert Pujols	154	592	133	196	46	123	84	52	.331
Scott Rolen	142	500	109	157	34	124	72	92	.314
Tony Womack	145	553	91	170	5	38	36	60	.307
Jim Edmonds	153	498	102	150	42	111	101	150	.301
Total	148.5	2143	435	673	127	396	293	354	.314



2004 Cardinal Batting Averages

Player	G	AB	R	H	HR	RBI	BB	K	AVG
Albert Pujols	154	592	133	196	46	123	84	52	.331
Scott Rolen	142	500	109	157	34	124	72	92	.314
Tony Womack	145	553	91	170	5	38	36	60	.307
Jim Edmonds	153	498	102	150	42	111	101	150	.301
Total	148.5	2143	435	673	127	396	293	354	.314

Structural part of data tables

There are usually four structural parts of a table: caption, head, foot (if the table has some summary figures), and body. These are coded as <caption>,<thead>,<tfoot> and <tbody>. These components are part of the HTML/XHTML specification, so it is certainly alright to use them, even though your page may validate and be accessible without some of them.

<caption>

The <caption> is just a title for your table, and is important for accessibility. When you use the <caption> tag, you explicitly associate the table's title with the table. Example:

- Bad: <p>2004 Cardinal Batting Averages</p><table><tr>...
- Good: <table><caption>2004 Cardinal Batting Averages </caption><tr>...

While the <caption> resides within the <table> tag, it usually does not share the table's default styles, like background or border. Of all the structural table tags, this one is most useful for accessibility since it tells the user what is in the table.

<thead>

<thead> identifies the column headers in your table. While not always necessary in a simple table, <thead> adds structural markup to your table and lets the browser know that the tags in the <thead> are important markers. If you have a complex table with multiple sections, you may have several different <thead> sections.

In our example above, the <thead> section looks like this:

```
<thead>
  <tr>
    <th scope=“col“>Player</th>
    <th scope=“col”><abbr title=“Games Played“>G</abbr></th>
    <th scope=“col”><abbr title=“At Bats“>AB</abbr></th>
    <th scope=“col”><abbr title=“Runs Scored“>R</abbr></th>
    <th scope=“col”><abbr title=“Hits“>H</abbr></th>
    <th scope=“col”><abbr title=“Home Runs“>HR</abbr></th>
    <th scope=“col”><abbr title=“Runs Batted In“>RBI</abbr></th>
    <th scope=“col”><abbr title=“Walks“>BB</abbr></th>
    <th scope=“col”><abbr title=“Strikeouts“>K</abbr></th>
    <th scope=“col”><abbr title=“Batting Average“>AVG</abbr></th>
  </tr>
</thead>
```

<tfoot>

<tfoot> comes, somewhat counter intuitively, before the body of the table. <tfoot> is a way of marking the summary data of a table, like the “totals” row (or rows). This allows browsers to display the data in <tfoot> before the rest of the page fully loads. <tfoot> is displayed at the bottom of the table.

Many simple tables may not have a “totals” row, and may not need to use a <tfoot>. Complex tables, however, may have one or more <tfoot> sections.

In our example, the <tfoot> section looks like this:

```
<tfoot>
  <tr>
    <th>Total</th>
    <td>148.5</td>
    <td>2143</td>
    <td>435</td>
    <td>673</td>
    <td>127</td>
    <td>396</td>
    <td>293</td>
    <td>354</td>
    <td>.314</td>
  </tr>
</tfoot>
```

<tbody>

<tbody> is structural markup to identify the main data items of the table and follows <tfoot>. With complex tables, it may be possible to have several <tbody> sections, although if you find yourself in that situation, you may want to consider breaking the data out into separate tables.

If you are doing any coding to generate your tables, using these tags will help your programmers identify each section of the table and make it easier to code.

Here's the <tbody>:

```
 <tbody>
   <tr>
     <th scope=”row”>Albert Pujols</th>
     <td>154</td>
     <td>592</td>
     <td>133</td>
     <td>196</td>
     <td>46</td>
     <td>123</td>
     <td>84</td>
     <td>52</td>
     <td>.331</td>
   </tr>
...
   <tr>
     <th scope=”row”>Jim Edmonds</th>
     <td>153</td>
     <td>498</td>
     <td>102</td>
     <td>150</td>
     <td>42</td>
     <td>111</td>
     <td>101</td>
     <td>150</td>
     <td>.301</td>
   </tr>
 </tbody>
```

Making the table accessible

Now that we've identified our table parts and put an accessible caption on the table, let's look at making all those columns and rows mean something.

Table headers

You've probably done something like this in the past in order to label the columns in your tables:

```
<tr>
  <td><center><b>Player</b></center></td>
  <td><center><b>Games</b></center></td>
  <td><center><b>At-bats</b></center></td>
  <td><center><b>Hits</b></center></td>
  <td><center><b>Home Runs</b></center></td>
  <td><center><b>RBI</b></center></td>
 </tr>
```

What's wrong with that? Well, there's nothing in the code to tell the browser or the user what the tags really mean. As far as the code goes, this is all data. The only way we would know that these are column headers is by context, and that is getting away from usability and accessibility.

So how should we do it? By using a table header <th>, like this:

```
  <thead>
        <tr>
    <th>Player</th>
    <th>G</th>
    <th>AB</th>
    <th>R</th>
    <th>H</th>
    <th>HR</th>
    <th>RBI</th>
    <th>BB</th>
    <th>K</th>
    <th>AVG</th>
  </tr>
</thead>
```

When we use <th>, we identify those cells as headers for the data that follows. Most browsers will automatically center and bold the <th> tag as well, so you don't have to add additional styling or tags.

Now that we've added headers, we need to tie those headers in to the data columns. We will do this with a “scope” attribute. This tells the browser that the cells within the scope of that header are related.

```
<th scope="col">Player</th>
```

will association every player name in that column with the caption “player”.

You can also use <th scope=”row”> on each line of data to identify a header for a row of data. That way, you create a cross-reference for each cell with a column and row.

Guideline: Use table headers <th>, captions <caption> and scope <th scope=”col”> to define data tables. Example:

```
<table>
<caption>Phone List</caption><tr>
<th scope="col">Last Name</th><th scope="col">First Name</th><th
scope="col">Number</th></tr>
```

```
<tr><td>Boeckman</td><td>Debbie</td><td>(573)  555-8400</td></tr>
<tr><td>Strange</td><td>Lanie</td><td>(573)  555-2500</td></tr>
</table>
```

More on tables

If you have more complex tables, such as tables with multiple sections or headers that span multiple rows, accessibility is a little more complicated. You can can find more information on tables at these websites:

- http://www.w3.org/TR/REC-html40/struct/tables.html
- http://www.w3.org/TR/2000/NOTE-WCAG10-HTML-TECHS-20000920/
- http://www.webaim.org/techniques/tables/
- http://www.mcu.org.uk/articles/tables.html
- http://www.yourhtmlsource.com/tables/tablesaccessibility.html
- http://www.ferg.org/section508/accessible_tables.html

Guideline: Do not incorporate these tags in layout table.

(i)Frames shall be titled with text that facilitates frame identification and navigation.

Guideline: **Avoid using frames**. Use a server-side includes to create headers, menus and other items that are common to each of your pages.

Frames are a method to include content from several sources in one screen on a browser. The most common elements are sidebars, headers and footers. These are all displayed in separate windows in frame in the browser window.

For the most part, any element you would normally put in a frame would be better used in a server-side include. This keeps all the elements in a single page instead of in separate pages in a single frame.

Frames can also cause confusion with navigation. When a user clicks on a link in a frame, the URL in the address bar does not change. This makes it very difficult to link to items in a frame.

Standard: If you have to use frames, each frame must be given a title attribute.

If the purpose of the frame is not apparent from the title, an alt or longdesc attribute should be included.

Standard: Use <noframe> to identify content for screen readers to read.

For specific techniques for making frames accessible, see the W3C guidelines at http://www.w3.org/TR/WCAG10-HTML-TECHS/#frames.

(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

Flickering pages or images are distracting and can trigger certain types of epileptic seizures. Avoid using Javascript or DHTML effects that make the page flicker. When possible, avoid animation.

Do not use the <blink> and <marquee> tags. In addition to being browser-specific, non-standard code, <marquee> is often read by a screen reader one letter at a time as it appears on the screen and <blink>, well <blink> blinks, which is the same as a flicker.

(k) A text-only page, with equivalent information or functionality, shall be provided to make a Web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.

If you have followed the standards and guidelines, you won't need to make a separate accessible text-only page. That's the whole idea behind standards.

If your page consists of image map navigation, for instance, you may have to create a separate page. If more than one of your pages are image maps, though, you will find that it takes a lot of work to maintain two versions of each page.

Guideline: If you have to create a separate text-only page, start thinking about how you can revise that page to make it accessible.

(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.

While server-side scripting poses no problem, client-side scripting, Javascript and event handlers may not be usable to screen readers, text browsers or browsers with scripting disabled.

This is particularly true of some drop-down menus, where the user does not have a “Go” button to click.

Testing

Disable scripting and refresh to see if you can still read and navigate the page. If you can't, come up with a <noscript> alternative to the script.

(m) When a Web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with the standards set forth under “Software Applications and Operating Systems”, paragraphs (a) through (l) contained within this document.

If you want to use PDFs, Flash, Shockwave, Word or any other plug-in, make sure you give the user a link to get the plug-in. Make sure the plug-in is accessible. If the application is not accessible, be sure to provide an alt-text or longdesc tag to describe the content. If you are using Flash, use <noembed>.

(n) When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

Use the <label> tag to associate form labels with their form fields. Example:

```
<p><label for="email">E-mail address: </label>
<input type="text" id="email"></p>
```

You can also use the <label> tag to wrap descriptive text with a form field. Example:

```
<p><label>First Name
<input type="text" name="firstname"></label>
<label> Last Name
<input type="text" name="lastname"></label></p>
```

Labels are especially important for check boxes and radio buttons, since they let the user select the label text to toggle the button. This is a great help for users with mobility difficulties.

Add tab order such as <input tabindex=”1”> to forms to make the fields flow more smoothly with keyboard use instead of a mouse.

Use <fieldset> and <legend> to separate different parts of your form.

Sample code for an accessible form:

```
<fieldset>
<legend>Policy Information</legend>
<label for="covtype">Coverage Type: </label>
<input type="text"  name="covtype" id="covtype" tabindex="1">
</select>
<label for="produc">Producer of Account: </label>
<input type="text" name="produc" id="produc" value="" tabindex="2" />
<label for="policy">Policy Number: </label>
<input type="text" name="policy" id="policy" value="" tabindex="3" />
</fieldset>
<fieldset>
<legend>Effective Date:</legend>
<label for="effmon">Month:</label>
<select name="effmon" id="effmon" tabindex="4">...
```

More information on forms is available at http://www.w3.org/TR/html401/interact/forms.html

If you have difficulties making your form accessible, provide a form which can be downloaded then mailed or e-mailed, or a phone number someone can call for assistance.

(o) A method shall be provided that permits users to skip repetitive

navigation links.

Since screen readers read every link on the page in order, most users will have to listen to all of your links before they hear any of your content. If your pages are like most, this means the user will hear all the link in your header and sidebar for each page.

Instead, use hidden links to allow the user to skip the menus and go straight to the content. Here's how:

```
...
</head>
<body>
<p class="none">[<a href="#content" title="Skip to content">Skip to content</a>]</p>

...[header][sidebar]...

<div id="content">
<p>This is the content part of the page.</p>
</div>
```

Your CSS would look like this:

```
.none  {
        width:0;
        height: 0;
        overflow: hidden;
        position:absolute;
}
```

Some people have suggested using “display: none; “ to hide the code, but this ends up hiding the code for any browser that reads CSS, like some screen readers. The method demonstrated above doesn't keep the text from being read, it just hides the visual display, so the link stays accessible. (A tip of the hat to DESE, who discovered this bit of CSS. Thanks, Lainie!)

Testing

Turn off the style sheet and see if you can skip your navigation elements to get to the content.

(p) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

Timed response may be used on login user sessions, forms, refresh and redirects. Users with disabilities generally take longer to navigate through a page, and may need more time.

Imagine taking a survey or quiz when suddenly the page auto-refreshes and wipes out all your answers. Or a slide show that goes on to the next slide before you finished reading the first one.

If you have a page that does an auto-refresh or a redirect, give enough time for a user to read your message and act (or not act). Consider a 10-second or longer pause, and make sure you put a link on the page to the redirect location so the user can click there if she just can't wait that long.

Example:

```
<meta http-equiv-"refresh" content="10" url="newpage.htm" />
...
<p>This page has moved to <a
        href="newpage.htm">http://oa.mo.gov.dmd/newpage.htm</a>.</p>
<p> You will be automatically redirected to that page in 10 seconds.</p>
```

(q) Contact information for issues related to accessibility shall be provided on each entry page.

This item is unique to Missouri. Your home page and main pages should have a link to contact someone at your agency regarding any accessibility issues. This link could go to the web master, accessibility officer, or any other designated person in your agency.

Other accessibility points

If you have followed the guidelines and standards in this manual, you'll recognize these guidelines. If these items are new to you, you didn't read closely enough back in the beginning. Start again.

Links

Avoid using the phrase "click here" in your links. Use descriptive text within your normal sentence structure for your links.

- Bad: "To review the web guidelines on-line, click here."
- Good: "Review the web guidelines on-line."

If you cannot find a suitable phrase in your text, you can add a title attribute to your link. This will show up as a "tool tip" box in a browser, and should be read aloud by a screen reader.

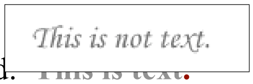
If you have a series of links that run together, like in a toolbar, make sure you use a space or a symbol to separate the links. Some older screen readers stumble over consecutive links without a separator.

You can hide the separator from a visual browser with CSS, using the same code example used in item (o) of the Missouri Accessibility Standards.

Image text

Guideline: You should avoid using graphics to display text.

This means that, instead of using Javascript rollover buttons for your navigation links in a sidebar, you should use text and CSS to display the link in its various states.

- Bad: 
This is not text.
- Good: This is text.

If you need to add text as part of a graphic, be sure that the text is available in an alt or longdesc attribute.

CSS rollovers are simple, as long as you remember to put the link styles in the right order:

- a:link
- a:visited
- a:hover
- a:active

ASCII art and emoticons

Punctuation and symbols add meaning to your documents, but when you use them outside of a sentence, they lose meaning and can cause confusion.

Graphics created with punctuation marks and text symbols have no context and can confuse screen readers. While ASCII art is usually found in e-mail signatures, it can also be found on the web.

Example:

```
    _    ____   ____ ___ ___      _         _   
   / \  / ___| / ___|_ _|_ _|    / \   _ __| |_ 
  / _ \ \___ \| |    | | | |    / _ \ | '__| __|
 / ___ \ ___) | |___ | | | |   / ___ \| |  | |_ 
/_/   \_\____/ \____|___|___| /_/   \_\_|   \__|
```

This example shows the word “ASCII Art” made up entirely of dashes, underscores, slashes and backslashes. While a sighted person may see a word there, a screen reader may see this as “hyphen space space hyphen hyphen hyphen space space, etc.”

Emoticons are characters that try to add meaning and nuance to a text message. The most popular example is the “smiley”, a colon, hyphen and closed parenthesis “:-)”. While these can add some context to an e-mail or text message (like, “Don't take this too seriously.”), an emoticon doesn't have any semantic meaning to a screen reader.

This really should not be much of a burden for a state web developer. Think about it. Do you really want to have content on your site that is so ambiguous that you have to tell the user that you are just kidding?

Resources

Section 508: http://www.section508.gov/
http://www.access-board.gov/sec508/508standards.htm

Accessibility book: “Building Accessible Websites” by Joe Clark
http://www.joeclark.org/book/

Accessible Tables: http://www.ferg.org/section508/accessible_tables.html

Dreamweaver Accessibility Extension:
http://www.usablenet.com/frontend/508as_entry.jsp

SMSU Accessibility Compliance Checkpoints:
http://wact.smsu.edu/Checkpoints/default.htm